

APPLICATION FOR UNITED STATES PATENT

in the name of

James Crawford

of

America Online, Inc.

for

Gradual Image Display

Fish & Richardson P.C.
601 Thirteenth Street, NW
Washington, DC 20005
Tel.: (202) 783-5070
Fax: (202) 783-2331

ATTORNEY DOCKET:

06975-098002

Gradual Image Display

This application is a continuation of U.S. Patent Application No. 09/632,805, filed August 4, 2000, which claims the benefit of U.S. Provisional Application No. 60/215,765 filed June 30, 2000.

TECHNICAL FIELD

5 The present invention relates generally to transferring files between subscribers of a communications system.

BACKGROUND

Online service providers are constantly offering new services and upgrading existing services to enhance their subscribers' online experience. Subscribers have virtually on-
10 demand access to news, weather, financial, sports, and entertainment services as well as the ability to transmit electronic messages and to participate in online discussion groups. For example, subscribers of online service providers such as America Online or CompuServe may view and retrieve information on a wide variety of topics from servers located throughout the world. A server may be maintained by the service provider or by a third party
15 provider who makes information and services available through the worldwide network of computers that make up the online service.

America Online has provided subscribers with the ability to send and receive instant messages. Instant messages are private online conversations between two or more people who have subscribed to the instant messaging service and have installed the necessary
20 software. Because such online conversations take place virtually in real time, instant messaging can provide immediate access to desired information. Instant messaging is becoming a preferred means of communicating among online subscribers.

SUMMARY

In one general aspect, an image sent between subscribers of a communications system
25 is displayed by receiving electronic data corresponding to the image; displaying a blurred view of the image; and gradually displaying a sharper version of the image in response to user input.

devices 120 and/or client controllers 125. For example, the client system 105 may include one or more general-purpose computers (e.g., personal computers), one or more special-purpose computers (e.g., devices specifically programmed to communicate with each other and/or the host system 110), or a combination of one or more general-purpose computers and
5 one or more special-purpose computers. The client system 105 may be arranged to operate within or in concert with one or more other systems, such as for example, one or more LANs ("Local Area Networks") and/or one or more WANs ("Wide Area Networks").

The client device 120 is generally capable of executing instructions under the command of a client controller 125. The client device 120 is connected to the client
10 controller 125 by a wired or wireless data pathway 130 capable of delivering data.

The client device 120 and client controller 125 each typically includes one or more hardware components and/or software components. An example of a client device 120 is a general-purpose computer (e.g., a personal computer) capable of responding to and executing instructions in a defined manner. Other examples include a special-purpose computer, a
15 workstation, a server, a device, a component, other equipment or some combination thereof capable of responding to and executing instructions. An example of client controller 125 is a software application loaded on the client device 120 for commanding and directing communications enabled by the client device 120. Other examples include a program, a
20 piece of code, an instruction, a device, a computer, a computer system, or a combination thereof, for independently or collectively instructing the client device 120 to interact and operate as described herein. The client controller 125 may be embodied permanently or temporarily in any type of machine, component, equipment, storage medium, or propagated
signal capable of providing instructions to the client device 120.

The communications link 115 typically includes a delivery network 160 making a
25 direct or indirect communication between the client system 105 and the host system 110, irrespective of physical separation. Examples of a delivery network 160 include the Internet, the World Wide Web, WANs, LANs, analog or digital wired and wireless telephone networks (e.g. PSTN, ISDN, or xDSL), radio, television, cable, satellite, and/ or any other delivery mechanism for carrying data. The communications link 115 may include communication
30 pathways 150, 155 that enable communications through the one or more delivery networks

160 described above. Each of the communication pathways 150, 155 may include, for example, a wired, wireless, cable or satellite communication pathway.

The host system 110 includes a host device 135 capable of executing instructions under the command and direction of a host controller 140. The host device 135 is connected
5 to the host controller 140 by a wired or wireless data pathway 145 capable of carrying and delivering data.

The host system 110 typically includes one or more host devices 135 and/or host controllers 140. For example, the host system 110 may include one or more general-purpose computers (e.g., personal computers), one or more special-purpose computers (e.g., devices
10 specifically programmed to communicate with each other and/or the client system 105), or a combination of one or more general-purpose computers and one or more special-purpose computers. The host system 110 may be arranged to operate within or in concert with one or more other systems, such as, for example, one or more LANs ("Local Area Networks") and/or one or more WANs ("Wide Area Networks").

15 The host device 135 and host controller 140 each typically includes one or more hardware components and/or software components. An example of a host device 135 is a general-purpose computer (e.g., a personal computer) capable of responding to and executing instructions in a defined manner. Other examples include a special-purpose computer, a workstation, a server, a device, a component, other equipment or some combination thereof
20 capable of responding to and executing instructions. An example of host controller 140 is a software application loaded on the host device 135 for commanding and directing communications enabled by the host device 135. Other examples include a program, a piece of code, an instruction, a device, a computer, a computer system, or a combination thereof, for independently or collectively instructing the host device 135 to interact and operate as
25 described herein. The host controller 140 may be embodied permanently or temporarily in any type of machine, component, equipment, storage medium, or propagated signal capable of providing instructions to the host device 135.

Fig. 2 illustrates a communication system 200 including a client system 205 communicating with a host system 210 through a communications link 215. Client system
30 205 typically includes one or more client devices 220 and one or more client controllers 225 for controlling the client devices 220. Host system 210 typically includes one or more host

devices 235 and one or more host controllers 240 for controlling the host devices 235. The communications link 215 may include communication pathways 250, 255 enabling communications through the one or more delivery networks 260.

Examples of each element within the communication system of Fig. 2 are broadly described above with respect to Fig. 1. In particular, the host system 210 and communications link 215 typically have attributes comparable to those described with respect to host system 110 and communications link 115 of Fig. 1. Likewise, the client system 205 of Fig. 2 typically has attributes comparable to and illustrates one possible embodiment of the client system 105 of Fig. 1.

The client device 220 typically includes a general purpose computer 270 having an internal or external storage 272 for storing data and programs such as an operating system 274 (e.g., DOS, Windows™, Windows 95™, Windows 98™, Windows 2000™, Windows NT™, OS/2, or Linux) and one or more application programs. Examples of application programs include authoring applications 276 (e.g., word processing, database programs, spreadsheet programs, or graphics programs) capable of generating documents or other electronic content; client applications 278 (e.g., AOL client, CompuServe client, AIM client, AOL TV client, or ISP client) capable of communicating with other computer users, accessing various computer resources, and viewing, creating, or otherwise manipulating electronic content; and browser applications 280 (e.g., Netscape's Navigator or Microsoft's Internet Explorer) capable of rendering standard Internet content.

The general-purpose computer 270 also includes a central processing unit 282 (CPU) for executing instructions in response to commands from the client controller 225. In one implementation, the client controller 225 includes one or more of the application programs installed on the internal or external storage 272 of the general-purpose computer 270. In another implementation, the client controller 225 includes application programs externally stored in and performed by one or more device(s) external to the general-purpose computer 270.

The general-purpose computer typically will include a communication device 284 for sending and receiving data. One example of the communication device 284 is a modem. Other examples include a transceiver, a set-top box, a communication card, a satellite dish, an antenna, or another network adapter capable of transmitting and receiving data over the

communications link 215 through a wired or wireless data pathway 250. The general-purpose computer 270 also may include a TV ("television") tuner 286 for receiving television programming in the form of broadcast, satellite, and/or cable TV signals. As a result, the client device 220 can selectively and/or simultaneously display network content received by communications device 284 and television programming content received by the TV tuner 286.

The general-purpose computer 270 typically will include an input/output interface 288 for wired or wireless connection to various peripheral devices 290. Examples of peripheral devices 290 include, but are not limited to, a mouse 291, a mobile phone 292, a personal digital assistant 293 (PDA), a keyboard 294, a display monitor 295 with or without a touch screen input, and/or a TV remote control 296 for receiving information from and rendering information to subscribers.

Although Fig. 2 illustrates devices such as a mobile telephone 292, a PDA 293, and a TV remote control 296 as being peripheral with respect to the general-purpose computer 270, in another implementation, such devices may themselves include the functionality of the general-purpose computer 270 and operate as the client device 220. For example, the mobile phone 292 or the PDA 293 may include computing and networking capabilities and function as a client device 220 by accessing the delivery network 260 and communicating with the host system 210. Furthermore, the client system 205 may include one, some or all of the components and devices described above.

Referring to Fig. 3, a communications system 300 is capable of delivering and exchanging information between a client system 305 and a host system 310 through a communication link 315. Client system 305 typically includes one or more client devices 320 and one or more client controllers 325 for controlling the client devices 320. Host system 310 typically includes one or more host devices 335 and one or more host controllers 340 for controlling the host devices 335. The communications link 315 may include communication pathways 350, 355 enabling communications through the one or more delivery networks 360.

Examples of each element within the communication system of Fig. 3 are broadly described above with respect to Figs. 1 and 2. In particular, the client system 305 and the communications link 315 typically have attributes comparable to those described with

respect to client systems 105 and 205 and communications links 115 and 215 of Figs. 1 and 2. Likewise, the host system 310 of Fig. 3 may have attributes comparable to and illustrates one possible embodiment of the host systems 110 and 210 shown in Figs. 1 and 2, respectively.

5 The host system 310 includes a host device 335 and a host controller 340. The host controller 340 is generally capable of transmitting instructions to any or all of the elements of the host device 335. For example, in one implementation, the host controller 340 includes one or more software applications loaded on the host device 335. However, in other implementations, as described above, the host controller 340 may include any of several
10 other programs, machines, and devices operating independently or collectively to control the host device 335.

 The host device 335 includes a login server 370 for enabling access by subscribers and routing communications between the client system 305 and other elements of the host device 335. The host device 335 also includes various host complexes such as the depicted
15 OSP ("Online Service Provider") host complex 380 and IM ("Instant Messaging") host complex 390. To enable access to these host complexes by subscribers, the client system 305 includes communication software, for example, an OSP client application and an IM client application. The OSP and IM communication software applications are designed to facilitate the subscriber's interactions with the respective services and, in particular, may provide
20 access to all the services available within the respective host complexes.

 Typically, the OSP host complex 380 supports different services, such as email, discussion groups, chat, news services, and Internet access. The OSP host complex 380 is generally designed with an architecture that enables the machines within the OSP host complex 380 to communicate with each other and employs certain protocols (i.e., standards,
25 formats, conventions, rules, and structures) to transfer data. The OSP host complex 380 ordinarily employs one or more OSP protocols and custom dialing engines to enable access by selected client applications. The OSP host complex 380 may define one or more specific protocols for each service based on a common, underlying proprietary protocol.

 The IM host complex 390 is generally independent of the OSP host complex 380, and
30 supports instant messaging services irrespective of a subscriber's network or Internet access. Thus, the IM host complex 390 allows subscribers to send and receive instant messages,

whether or not they have access to any particular ISP. The IM host complex 390 may support associated services, such as administrative matters, advertising, directory services, chat, and interest groups related to the instant messaging. The IM host complex 390 has an architecture that enables all of the machines within the IM host complex to communicate with each other. To transfer data, the IM host complex 390 employs one or more standard or exclusive IM protocols.

The host device 335 may include one or more gateways that connect and therefore link complexes, such as the OSP host complex gateway 385 and the IM host complex gateway 395. The OSP host complex gateway 385 and the IM host complex 395 gateway may directly or indirectly link the OSP host complex 380 with the IM host complex 390 through a wired or wireless pathway. Ordinarily, when used to facilitate a link between complexes, the OSP host complex gateway 385 and the IM host complex gateway 395 are privy to information regarding the protocol type anticipated by a destination complex, which enables any necessary protocol conversion to be performed incident to the transfer of data from one complex to another. For instance, the OSP host complex 380 and IM host complex 390 generally use different protocols such that transferring data between the complexes requires protocol conversion by or at the request of the OSP host complex gateway 385 and/or the IM host complex gateway 395.

Referring to Fig. 4, a communications system 400 is capable of delivering and exchanging information between a client system 405 and a host system 410 through a communication link 415. Client system 405 typically includes one or more client devices 420 and one or more client controllers 425 for controlling the client devices 420. Host system 410 typically includes one or more host devices 435 and one or more host controllers 440 for controlling the host devices 435. The communications link 415 may include communication pathways 450, 455 enabling communications through the one or more delivery networks 460. As shown, the client system 405 may access the Internet 465 through the host system 410.

Examples of each element within the communication system of Fig. 4 are broadly described above with respect to Figs. 1-3. In particular, the client system 405 and the communications link 415 typically have attributes comparable to those described with respect to client systems 105, 205, and 305 and communications links 115, 215, and 315 of

Figs. 1-3. Likewise, the host system 410 of Fig. 4 may have attributes comparable to and illustrates one possible embodiment of the host systems 110, 210, and 310 shown in Figs. 1-3, respectively. However, Fig. 4 describes an aspect of the host system 410, focusing primarily on one particular implementation of OSP host complex 480. For purposes of communicating with an OSP host complex 480, the delivery network 460 is generally a telephone network.

The client system 405 includes a client device 420 and a client controller 425. The client controller 425 is generally capable of establishing a connection to the host system 410, including the OSP host complex 480, the IM host complex 490 and/or the Internet 465. In one implementation, the client controller 425 includes an OSP application for communicating with servers in the OSP host complex 480 using exclusive OSP protocols. The client controller 425 also may include applications, such as an IM client application, and/or an Internet browser application, for communicating with the IM host complex 490 and the Internet 465.

The host system 410 includes a host device 435 and a host controller 440. The host controller 440 is generally capable of transmitting instructions to any or all of the elements of the host device 435. For example, in one implementation, the host controller 440 includes one or more software applications loaded on one or more elements of the host device 435. However, in other implementations, as described above, the host controller 440 may include any of several other programs, machines, and devices operating independently or collectively to control the host device 435.

The host system 410 includes a login server 470 capable of enabling communications with and authorizing access by client systems 405 to various elements of the host system 410, including an OSP host complex 480 and an IM host complex 490. The login server 470 may implement one or more authorization procedures to enable simultaneous access to the OSP host complex 480 and the IM host complex 490. The OSP host complex 480 and the IM host complex 490 are connected through one or more OSP host complex gateways 485 and one or more IM host complex gateways 495. Each OSP host complex gateway 485 and IM host complex gateway 495 may perform any protocol conversions necessary to enable communication between the OSP host complex 480, the IM host complex 490, and the Internet 465.

The OSP host complex 480 supports a set of services from one or more servers located internal to and external from the OSP host complex 480. Servers external to the OSP host complex 480 generally communicate through the Internet 465. Servers internal to the OSP complex 480 may be arranged in one or more configurations. For example, servers may
5 be arranged in large centralized clusters known as farms 4802 or in localized clusters known as pods 4804.

Farms 4802 are groups of servers located at centralized locations within the OSP host complex 480. Farms 4802 generally are dedicated to providing particular functionality and services to subscribers and clients from a centralized location, regardless of the location of
10 the subscriber or client. Farms 4802 are particularly useful for providing services that depend upon other processes and services for information, such as, for example, chat, email, instant messaging, news, newsgroups, search, stock updates, and weather. Thus, farms 4802 tend to rely on connections with external resources such as the Internet 465 and/or other servers within the OSP host complex 480.

To reduce the time delays and congestion inherent in centralized processing, some
15 services offered by the OSP host complex 480 are provided from localized servers, generally known as pods 4804. Each pod 4804 includes one or more interrelated servers capable of operating together to provide one or more services offered by the OSP host complex 480 in a geographically localized manner, the servers within a pod 4804 generally operating
20 independently rather than relying on resources external to the pod 4804 to operate. A pod 4804 may cache content received from external sources, such as farms 4802 or the Internet 465, making frequently requested information readily available to local subscribers served by the pod 4804. In this way, pods 4804 are particularly useful in providing services that are independent of other processes and servers such as, for example, routing, keywords, and
25 downloading certain software and graphical interface updates with reduced processing time and congestion. The determination of which servers and processes are located in the pod 4804 is made by the OSP according to load distribution, frequency of requests, demographics, and other factors.

In addition to farms 4802 and pods 4804, the implementation of Fig. 4 also includes
30 one or more non-podded servers 4806. In general, the non-podded server 4806 may be dedicated to performing a particular service that relies on other processes and services for

information and may be directly or indirectly connected to resources outside of the OSP host complex 480, such as the Internet 465 and the IM host complex 490, through an OSP gateway 4808. In the event that subscriber usage of the particular service is relatively high, the non-podded server 4806 may be included in a farm.

5 In the implementation of Fig. 4, a pod 4810, shown in more detail, includes a routing processor 4812. In a packet-based implementation, the client system 405 may generate information requests, convert the requests into data packets, sequence the data packets, perform error checking and other packet-switching techniques, and transmit the data packets to the routing processor 4812. Upon receiving data packets from the client system 405, the
10 routing processor 4812 may directly or indirectly route the data packets to a specified destination within or outside of the OSP host complex 480. In general, the routing processor 4812 will examine an address field of a data request, use a mapping table to determine the appropriate destination for the data request, and direct the data request to the appropriate destination.

15 For example, in the event that a data request from the client system 405 can be satisfied locally, the routing processor 4812 may direct the data request to a local server 4814 in the pod 4810. In the event that the data request cannot be satisfied locally, the routing processor 4812 may direct the data request internally to one or more farms 4802, one or more other pods 4804, or one or more non-podded servers 4806 in the OSP host complex 480 or
20 may direct the data request externally to the Internet 465 or the IM host complex 490 through an OSP/pod gateway 4816.

The routing processor 4812 also may direct data requests and/or otherwise facilitate communication between the client system 405 and the Internet 465. In one implementation, the client system 405 uses an OSP client application to convert standard Internet content and
25 protocols into OSP protocols and vice versa. For example, when a browser application transmits a request in standard Internet protocol, the OSP client application can intercept the request, convert the request into an OSP protocol and send the converted request to the routing processor 4812 in the OSP host complex 480. The routing processor 4812 recognizes the Internet 465 as the destination and routes the data packets to an IP ("Internet Protocol")
30 tunnel 4818. The IP tunnel 4818 converts the data from the OSP protocol back into standard Internet protocol and transmits the data to the Internet 465. The IP tunnel 4818 also converts

the data received from the Internet in the standard Internet protocol back into the OSP protocol and sends the data to the routing processor 4812 for delivery back to the client system 405. At the client system 405, the OSP client application converts the data in the OSP protocol back into standard Internet content for communication with the browser application.

5 The IP tunnel 4818 may act as a buffer between the client system 405 and the Internet 465, and may implement content filtering and time saving techniques. For example, the IP tunnel 4818 can check parental controls settings of the client system 405 and request and transmit content from the Internet 465 according to the parental control settings. In addition, the IP tunnel 4818 may include a number a caches for storing frequently accessed
10 information. If requested data is determined to be stored in the caches, the IP tunnel 4818 may send the information to the client system 405 from the caches and avoid the need to access the Internet 465.

In another implementation, the client system 405 may use standard Internet protocols and formatting to access the pod 4810 and the Internet 465. For example, the subscriber can
15 use an OSP TV client application having an embedded browser application installed on the client system 405 to generate a request in standard Internet protocol, such as HTTP ("HyperText Transport Protocol"). In a packet-based implementation, data packets may be encapsulated inside a standard Internet tunneling protocol, such as, for example, UDP ("User Datagram Protocol") and routed to a web tunnel 4820. The web tunnel 4820 may be a L2TP
20 ("Layer Two Tunneling Protocol") tunnel capable of establishing a point-to-point protocol (PPP) session with the client system 405. The web tunnel 4820 provides a gateway to the routing processor 4812 within the pod 4810, the Internet 465, and a web proxy 4822.

The web proxy 4822 can look up subscriber information from the IP address of the client system 405 to determine the subscriber's parental controls settings and other
25 demographic information. In this way, the web proxy 4822 can tailor the subscriber's content and user interfaces. The web proxy 4822 can also perform caching functions to store certain URLs ("Uniform Resource Locators") and other electronic content so that the web proxy 4822 can locally deliver information to the client system 405 and avoid the need to access the Internet 465 in the event that data requested by the client system 405 has been cached.

30 Referring to Fig. 5, a communications system 500 is capable of delivering and exchanging information between a client system 505 and a host system 510 through a

communication link 515. Client system 505 typically includes one or more client devices 520 and one or more client controllers 525 for controlling the client devices 520. Host system 510 typically includes one or more host devices 535 and one or more host controllers 540 for controlling the host devices 535. The communications link 515 may include
5 communication pathways 550, 555 enabling communications through the one or more delivery networks 560. As shown, the client system 505 may access the Internet 565 through the host system 510.

Examples of each element within the communication system of Fig. 5 are broadly described above with respect to Figs. 1-4. In particular, the client system 505 and the
10 communications link 515 typically have attributes comparable to those described with respect to client systems 105, 205, 305, and 405 and communications links 115, 215, 315, and 415 of Figs. 1-4. Likewise, the host system 510 of Fig. 5 may have attributes comparable to and illustrates one possible embodiment of the host systems 110, 210, 310, and 410 shown in Figs. 1-4, respectively. However, Fig. 5 describes an aspect of the host
15 system 510, focusing primarily on one particular implementation of IM host complex 590. For purposes of communicating with the IM host complex 590, the delivery network 560 is generally a telephone network.

The client system 505 includes a client device 520 and a client controller 525. The client controller 525 is generally capable of establishing a connection to the host system 510,
20 including the OSP host complex 580, the IM host complex 590 and/or the Internet 565. In one implementation, the client controller 525 includes an IM application for communicating with servers in the IM host complex 590 utilizing exclusive IM protocols. The client controller 525 also may include applications, such as an OSP client application, and/or an Internet browser application for communicating with the OSP host complex 580 and the
25 Internet 565, respectively.

The host system 510 includes a host device 535 and a host controller 540. The host controller 540 is generally capable of transmitting instructions to any or all of the elements of the host device 535. For example, in one implementation, the host controller 540 includes one or more software applications loaded on one or more elements of the host device 535.
30 However, in other implementations, as described above, the host controller 540 may include

any of several other programs, machines, and devices operating independently or collectively to control the host device 535.

The host system 510 includes a login server 570 capable of enabling communications with and authorizing access by client systems 505 to various elements of the host system 510, including an OSP host complex 580 and an IM host complex 590. The login server 570 may implement one or more authorization procedures to enable simultaneous access to the OSP host complex 580 and the IM host complex 590. The OSP host complex 580 and the IM host complex 590 are connected through one or more OSP host complex gateways 585 and one or more IM host complex gateways 595. Each OSP host complex gateway 585 and IM host complex gateway 595 may perform any protocol conversions necessary to enable communication between the OSP host complex 580, the IM host complex 590, and the Internet 565.

To access the IM host complex 590 to begin an instant messaging session, the client system 505 establishes a connection to the login server 570. The login server 570 typically determines whether the particular subscriber is authorized to access the IM host complex 590 by verifying a subscriber identification and password. If the subscriber is authorized to access the IM host complex 590, the login server 570 employs a hashing technique on the subscriber's screen name to identify a particular IM server 5902 for use during the subscriber's session. The login server 570 provides the client system 505 with the IP address of the particular IM server 5902, gives the client system 505 an encrypted key (i.e., a cookie), and breaks the connection. The client system 505 then uses the IP address to establish a connection to the particular IM server 5902 through the communications link 515, and obtains access to that IM server 5902 using the encrypted key. Typically, the client system 505 will be equipped with a Winsock API ("Application Programming Interface") that enables the client system 505 to establish an open TCP connection to the IM server 5902.

Once a connection to the IM server 5902 has been established, the client system 505 may directly or indirectly transmit data to and access content from the IM server 5902 and one or more associated domain servers 5904. The IM server 5902 supports the fundamental instant messaging services and the domain servers 5904 may support associated services, such as, for example, administrative matters, directory services, chat and interest groups. In general, the purpose of the domain servers 5904 is to lighten the load placed on the IM server

5902 by assuming responsibility for some of the services within the IM host complex 590. By accessing the IM server 5902 and/or the domain server 5904, a subscriber can use the IM client application to view whether particular subscribers ("buddies") are online, exchange instant messages with particular subscribers, participate in group chat rooms, trade files such as pictures, invitations or documents, find other subscribers with similar interests, get customized news and stock quotes, and search the Web.

In the implementation of Fig. 5, the IM server 5902 is directly or indirectly connected to a routing gateway 5906. The routing gateway 5906 facilitates the connection between the IM server 5902 and one or more alert multiplexors 5908, for example, by serving as a link minimization tool or hub to connect several IM servers to several alert multiplexors. In general, an alert multiplexor 5908 maintains a record of alerts and subscribers registered to receive the alerts.

Once the client system 505 is connected to the alert multiplexor 5908, a subscriber can register for and/or receive one or more types of alerts. The connection pathway between the client system 505 and the alert multiplexor 5908 is determined by employing another hashing technique at the IM server 5902 to identify the particular alert multiplexor 5908 to be used for the subscriber's session. Once the particular multiplexor 5908 has been identified, the IM server 5902 provides the client system 505 with the IP address of the particular alert multiplexor 5908 and gives the client system 505 an encrypted key (i.e., a cookie). The client system 505 then uses the IP address to connect to the particular alert multiplexor 5908 through the communication link 515 and obtains access to the alert multiplexor 5908 using the encrypted key.

The alert multiplexor 5908 is connected to an alert gate 5910 that, like the IM host complex gateway 595, is capable of performing the necessary protocol conversions to form a bridge to the OSP host complex 580. The alert gate 5910 is the interface between the IM host complex 590 and the physical servers, such as servers in the OSP host complex 580, where state changes are occurring. In general, the information regarding state changes will be gathered and used by the IM host complex 590. However, the alert multiplexor 5908 also may communicate with the OSP host complex 580 through the IM gateway 595, for example, to provide the servers and subscribers of the OSP host complex 580 with certain information gathered from the alert gate 5910.

The alert gate 5910 can detect an alert feed corresponding to a particular type of alert. The alert gate 5910 may include a piece of code (alert receive code) capable of interacting with another piece of code (alert broadcast code) on the physical server where a state change occurs. In general, the alert receive code installed on the alert gate 5910 instructs the alert
5 broadcast code installed on the physical server to send an alert feed to the alert gate 5910 upon the occurrence of a particular state change. Upon detecting an alert feed, the alert gate 5910 contacts the alert multiplexor 5908, which in turn, informs the client system 505 of the detected alert feed.

In the implementation of Fig. 5, the IM host complex 590 also includes a subscriber
10 profile server 5912 connected to a database 5914 for storing large amounts of subscriber profile data. The subscriber profile server 5912 may be used to enter, retrieve, edit, manipulate, or otherwise process subscriber profile data. In one implementation, a subscriber's profile data includes, for example, the subscriber's buddy list, alert preferences, designated stocks, identified interests, and geographic location. The subscriber may enter,
15 edit and/or delete profile data using an installed IM client application on the client system 505 to interact with the subscriber profile server 5912.

Because the subscriber's data is stored in the IM host complex 590, the subscriber does not have to reenter or update such information in the event that the subscriber accesses the IM host complex 590 using new or a different client system 505. Accordingly, when a
20 subscriber accesses the IM host complex 590, the IM server 5902 can instruct the subscriber profile server 5912 to retrieve the subscriber's profile data from the database 5914 and to provide, for example, the subscriber's buddy list to the IM server 5902 and the subscriber's alert preferences to the alert multiplexor 5908. The subscriber profile server 5912 also may communicate with other servers in the OSP host complex 590 to share subscriber profile data
25 with other services. Alternatively, user profile data may be saved locally on the client device 505.

Referring to Fig. 6, a communications system 600 is capable of delivering and exchanging information between a client system 605 and a host system 610 through a communication link 615. Client system 605 typically includes one or more client devices
30 620 and one or more client controllers 625 for controlling the client devices 620. Host system 610 typically includes one or more host devices 635 and one or more host controllers

640 for controlling the host devices 635. The communications link 615 may include communication pathways 650, 655 enabling communications through the one or more delivery networks 660.

5 Examples of each element within the communication system of Fig. 6 are broadly described above with respect to Figs. 1-5. In particular, the client system 605 and the communications link 615 typically have attributes comparable to those described with respect to client systems 105, 205, 305, 405 and 505 and communications links 115, 215, 315, 415 and 515 of Figs. 1-5. Likewise, the host system 610 of Fig. 6 may have attributes comparable to and illustrates one possible embodiment of the host systems 110, 210, 310, 410 and 510 shown in Figs. 1-5, respectively. However, Fig. 6 describes an aspect of the host system 610, focusing primarily on one particular implementation of IM host complex 690.

10 The client system 605 includes a client device 620 and a client controller 625. The client controller 625 is generally capable of establishing a connection to the host system 610, including the IM host complex 690. In one implementation, the client controller 625 includes an IM application for communicating with servers in the IM host complex 690 utilizing exclusive IM protocols.

15 The host system 610 includes a host device 635 and a host controller 640. The host controller 640 is generally capable of transmitting instructions to any or all of the elements of the host device 635. For example, in one implementation, the host controller 640 includes one or more software applications loaded on one or more elements of the host device 635. However, in other implementations, as described above, the host controller 640 may include any of several other programs, machines, and devices operating independently or collectively to control the host device 635.

20 The host system 610 includes a login server 670 capable of enabling communications with and authorizing access by client systems 605 to various elements of the host system 610, including the IM host complex 690. The IM host complex 690 includes an IM server network 6902 and an alert multiplexor network 6908. The IM server network 6902 is an interconnected network of IM servers and the alert multiplexor network 6908 is an interconnected network of alert multiplexors. Each IM server and each alert multiplexor can directly or indirectly communicate and exchange information with all of the IM servers in the IM server network 6902 and all of the alert multiplexors in the alert multiplexor network

25

30

6908. Each of the alert multiplexors in the alert multiplexor network 6908 is connected to several alert gates 6910 that receive different types of alerts. In the implementation of Fig. 6, the IM server network 6902 and the alert multiplexor network 6908 are interconnected by a routing gateway 6906 that serves as a common hub to reduce the number of connections.

5 A subscriber typically will be assigned to one IM server in the IM server network 6902 and to one alert multiplexor in the alert multiplexor network 6908 during a session based on one or more hashing techniques. However, the IM servers and the alert multiplexors are capable of storing subscriber information and other electronic content that may be accessed by the other IM servers and alert multiplexors. In one implementation, for example,
10 each IM server in the IM server network 6902 may be dedicated to serving a particular set of registered subscribers. Because all of the IM servers can communicate with each other, all subscribers can communicate with each other through instant messaging. In another implementation, each alert multiplexor in the alert multiplexor network 6908 may be dedicated to storing information about a particular set or subset of alerts. Because all of the
15 alert multiplexors can communicate with each other, all registered subscribers can receive all types of alerts. This networking arrangement enables the load to be distributed among the various servers in the IM host complex 690 while still enabling a subscriber to communicate, share information, or otherwise interact with other subscribers and servers in the IM host complex 690.

20 Referring to Fig. 7, a first client 702a, a second client 702b, and a host 704 interact according to a procedure 700 to enable and gradually display "buddy icons." Buddy icons are images such as cartoon characters, symbols, pictures, and other graphical images that subscribers can use to represent themselves or other buddies in their Buddy List[®]. In one implementation, buddy icons appear in the lower-left corner of instant messaging windows
25 when subscribers send messages back and forth to each other.

 The procedure 700 may be implemented by any suitable type of hardware, software, device, computer, computer system, equipment, component, program, application, code, storage medium, or propagated signal.

 Examples of each element of Fig. 7 are broadly described above with respect to Figs.
30 1-6. In particular, the first client 702a and the second client 702b typically have attributes comparable to those described with respect to client devices 120, 220, 320, 420, 520 and 620

and/or client controllers 125, 225, 325, 425, 525 and 625. The host 704 typically has attributes comparable to those described with respect to host device 135, 235, 335, 435, 535 and 635 and/or host controllers 140, 240, 340, 440, 540 and 640. The first client 702a, the second client 702b, and/or the host 704 may be directly or indirectly interconnected through
5 a known or described delivery network.

Each of the clients 702a, 702b is associated with a subscriber. To enable buddy icons, each subscriber selects a buddy icon and then sets certain transfer preferences for permitting buddy icons to be transferred to and from other subscribers. For example, a first subscriber may set transfer preferences governing which screen names or subscribers can send buddy
10 icons to client 702a. Likewise, a second subscriber may set transfer preferences governing which screen names or subscribers can send buddy icons to client 702b. Typically, each subscriber will be presented with a graphical UI ("User Interface") that permits the subscriber to select a buddy icon and set various transfer preferences. A subscriber's transfer preferences may be maintained locally or on the host 704.

Initially, the first subscriber selects a graphical image (step 710). The graphical
15 image may be selected from any of the images displayed by a browser when connecting to an OSP or the Internet. The first subscriber then installs the graphics file associated with the graphical image on the client 702a (step 715). In one implementation, the first subscriber saves the graphics file in a directory specifically designated for buddy icons on the client
20 702a. The first subscriber designates the selected graphical image as a buddy icon for the first subscriber (step 720a). Typically, the first subscriber will be presented with a UI to perform this designation.

The IM client application will size the graphical image to conform to standard buddy icon size. In one implementation, graphical images are standardized to 48x48 pixels.
25 Formats for the graphics files include, but are not limited to .BMP (bit-mapped), .GIF (graphics interchange format), .JPEG (joint photographic experts group), .ICO (icon), and .XBM (X bit map) file formats. In general, there is no limit to the height and width of the graphical image designated as a buddy icon.

There may, however, be a disk space limitation for graphical images designated as
30 buddy icons. In one implementation, the image cannot use more than 7168 bytes of disk space due to the limited capacity of the connection between the clients 702a, 702b and the

host 704. The limited capacity of the connection is designed to prevent objectionable images from being sent to subscribers through the IM host since most images require more than 8 kilobytes of disk space. The rationale is that the real-time quality of instant messaging affords little opportunity to monitor or screen images sent between subscribers. Therefore, it is safer for subscribers if the host 704 prevents images from being sent through the IM host. As described above, images may be transferred between subscribers only when the clients are directly connected through a socket connection, for example. Typically, subscribers will allow only trusted persons to directly connect to their computers.

Even though the host 704 prevents images from being sent in an instant message dialogue box, there is still the danger that a subscriber will choose a buddy icon that other subscribers find objectionable. To reduce the danger of a subscriber viewing an objectionable image, the client 702b gradually displays the buddy icon sent from another subscriber and affords the opportunity to accept or reject the buddy icon. This process will be described in more detail below.

At the second client 702b, the second subscriber selects buddy icon preferences governing which screen names or subscribers can send buddy icons to the client 702b (step 720b). The second subscriber may choose to never display buddy icons. When this option is set, the subscriber will not be able to see any buddy icons that have been chosen for other buddies. The second subscriber may select to never accept buddy icons. When this option is selected, the second subscriber is still able to see buddy icons that the second subscriber has chosen for other buddies. The second subscriber may select to accept buddy icons from users on the second subscribers Buddy List[®]. When this option is selected, the second subscriber will accept buddy icons from only familiar buddies that have been added to the subscribers Buddy List[®]. The second subscriber also may display a buddy icon knock-knock dialogue. When the option is selected, the second subscriber is alerted when a buddy desires to send a buddy icon. When this option is disabled, the second subscriber automatically receives buddy icons from users without any warning. In the procedure 700, it is assumed that the second subscriber has selected to display the buddy icon knock-knock dialogue.

After the first subscriber associated with the first client 702a has designated a graphical image as a buddy icon (step 720a), the first subscriber uses the first client 702a to communicate with the second subscriber. In particular, the first subscriber uses the first

client 702a to send an instant message addressed to the second subscriber along with image data corresponding to the buddy icon (step 725). The host 704 receives the message addressed to the second subscriber along with the image data (step 730). In one implementation, the host 704 is an IM host. The host 704 then authenticates the message addressed to the second subscriber (step 735). Typically, authentication is accomplished by confirming that the sender and the addressee of the message are registered subscribers of the host 704. Once the message has been authenticated, the host 704 sends the message and the image data to the second subscriber (step 740).

The client 702b receives the message and the image data associated with the buddy icon of the first subscriber from the host 704 (step 745). The client 702b then displays a blurred view of the buddy icon (step 750a) and also displays the message sent from the first subscriber (step 750b). In one implementation, the blurred view of the buddy icon is displayed in a knock-knock dialogue box. The knock-knock dialogue box may be presented to the second subscriber prior to the message from the first subscriber. Alternatively, the knock-knock dialogue box may overlay the message dialogue box.

When presented with the blurred view of the buddy icon, the second subscriber uses the second client 702b to accept or reject the buddy icon (step 755). The second subscriber may immediately accept the buddy icon from the first subscriber. In this case, a clear view of the buddy icon is rendered in the instant message dialogue box. The second subscriber also may immediately reject the buddy icon. In this case, the buddy icon is not presented to the second subscriber.

In some cases, however, the second subscriber may be unsure whether to accept or reject the buddy icon. In this case, the second subscriber may sharpen the view of the buddy icon (step 760). When the view of the buddy icon is sharpened, the second subscriber is presented with a better view of the buddy icon and can therefore make a more informed decision as to whether to accept or reject the buddy icon. After the view of the buddy icon is sharpened (step 760), the second subscriber again can choose whether to accept or reject the buddy icon (step 755).

In one implementation, the buddy icon is a 48x48 pixel image. When the second client 702b receives the image data from the host 704, the second client 702b displays every eighth pixel to the second subscriber. When the second subscriber selects to sharpen the

view of the buddy icon, the second client 702b displays every fourth pixel to the second subscriber. If the second subscriber chooses to sharpen the view of the buddy icon again, the second client 702b displays every other pixel to the second subscriber. Finally, if the second subscriber again sharpens the view of the buddy icon, the second client 702b displays the clear view of the buddy icon to the second subscriber. If at any time, the second subscriber suspects that the buddy icon is objectionable, the second subscriber has the option to reject the buddy icon. By presenting the second subscriber with a very poor resolution image and then gradually improving the resolution of the image, the second subscriber will be less likely to view an objectionable image.

In general, the client 702a and the client 702b communicate over an open connection, such as an open TCP connection established through the host 704. Typically, both clients 702a, 702b include a Winsock API for establishing an open TCP connection to the host 704 and a client application for accessing the host 704. The client devices 702a, 702b connect to the host 704 to establish the connection.

The clients 702a, 702b can use the connection to communicate with the host 704 and with each other. The connection remains open during the time that the first client 702a and the second client 702b are accessing the host 704. To access the host 704, each client 702a, 702b sends a request to the host 704. The requests identify the associated subscribers to the host 704 and to other subscribers using the subscribers' unique screen names. The host 704 verifies a subscriber's information (e.g., screen name and password) against data stored in a subscriber database. If the subscriber's information is verified, the host 704 authorizes access. If the subscriber's information is not verified, the host 704 denies access and sends an error message.

Upon accessing the host 704, the client 702a receives a list of the first subscriber's "buddies" that are currently online (i.e., accessing the host 704). Buddies are subscribers or screen names designated for exchanging instant messages. In general, the host 704 informs the first subscriber as to whether designated buddies are online or offline. The host 704 also informs any subscriber that has designated the first subscriber as a buddy that the first subscriber is online. The first subscriber can use an application running on the client 702a ("the client application") to view the online status of particular buddies, exchange instant messages with online buddies, participate in group chat rooms, trade files such as pictures,

invitations or documents, find other subscribers with similar interests, get customized news and stock quotes, and search the Web. Additionally, the subscriber can use the client application to transfer one or more files to or from the client device of another subscriber.

In one implementation, the client 702b attempts to establish a direct socket
5 connection (e.g., a peer-to-peer socket connection) to the client 702a using the IP address of the client 702a. In some circumstances, however, the client 702b may be unable to establish a direct socket connection to the client 702a, such as, for example, when the client 702a is behind a firewall. In the event that the client 702b cannot establish a direct socket
10 connection to the client 702a after a predetermined time period, the client 702b sends a connect message to the client 702a through the host 704. The connect message includes, for example, the message type, the screen name of the first subscriber, the screen name of the second subscriber, the IP address of the client 702b, and a randomly generated security number. The host 704 authenticates that the connect message from the client 702b is from a valid subscriber and then sends the connect message to the client 702a.

15 Upon receiving the connect message, the client 702a attempts to establish a direct socket connection to the client 702b of the second subscriber using the IP address of the client 702b. If, however, the client 702a is unable to establish a direct socket connection to the client 702b, such as, for example, when the client 702b is behind a firewall, the client 702a displays a message to that effect and sends an error message to the client 702b through
20 the host 704. The host 704 authenticates that the error message from the client 702a is from a valid subscriber and then sends the error message to the client 702b of the second subscriber. In response to the received error message, the client 702b displays a message to the subscriber.

If a direct socket connection has been established between the client 702a and the
25 client 702b, the client 702b verifies that the client 702a includes a valid client application by, for example, verifying the security number of the client application.

The direct connection bypasses the connection between the client 702a and the host 704. In one implementation, the host 704 limits the capacity of the connection to the client 702a. As a result, the client 702a cannot transfer files larger than a threshold file size (e.g.
30 8000 characters or bytes) through the host 704. In contrast to the capacity of the connection between the client 702a and the host 704, the capacity of the direct socket connection

between the client 702a and the client 702b is not limited. The client 702a, therefore, may transfer relatively large files (e.g. graphics files, executable files) to the client 702b.

Figs. 8-11 illustrate examples of graphical user interfaces (“UIs”) that may be presented to subscribers. In general, a graphical UI will be rendered on a subscriber’s client
5 device.

Referring to Fig. 8, a UI 800 includes a Buddy Icon preference box 805. A subscriber may use the Buddy Icon preference box 805 to set transfer preferences for buddy icons as well as to designate graphical images to be used as buddy icons. In one implementation, the Buddy Icon preference box 805 is used to select a subscriber’s own buddy icon to be
10 presented to other subscribers.

Referring to Fig. 9, a UI 900 includes a Buddy Icon preference box 905. In one implementation, the Buddy Icon preference box 905 is used to designate a buddy icon for a selected buddy.

Referring to Figs. 10A-10D, UI’s 1000A-1000D include Accept Buddy Icon dialogue
15 boxes 1005A-1005D. The Accept Buddy Icon dialogue boxes 1005A-1005D illustrate the progression of knock-knock dialogue boxes presented to a subscriber. That is, accept buddy icon dialogue box 1005B results from selecting the sharpen button in Accept Buddy Icon dialogue box 1005A. Likewise, Accept Buddy Icon dialogue box 1005C results from selecting the sharpened button in Accept Buddy Icon dialogue box 1005B. Finally, a clear
20 view of the buddy icon image is presented in Accept Buddy Icon dialogue box 1005D from selecting the sharpen button in Accept Buddy Icon dialogue box 1005C.

Referring to Fig. 11, a UI 1100 includes an instant message dialogue box 1105. In one implementation, the instant message dialogue box 1105 includes a buddy icon 1110. The buddy icon 1110 is rendered in the instant message dialogue box 1105 when designated as
25 the buddy icon for the selected buddy and/or when the buddy icon is selected by a buddy and accepted by the subscriber.

Although the examples above are described in conjunction with an instant messaging system, aspects may be applicable to other forms of communication, such as e-mail. Other embodiments are within the scope of the following claims.